

## IN THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

### Listing of Claims:

#### What is Claimed is:

1. (currently amended) In a Montgomery multiplication algorithm, a method of computing an output  $xyR^{-1} \bmod m$  from input integers  $m = (m_{n-1} \dots m_1 m_0)_b$ ,  $x = (x_{n-1} \dots x_1 x_0)_b$ ,  $y = (y_{n-1} \dots y_1 y_0)_b$ , with  $0 \leq x, y < m$ ,  $R = b^n$  with  $\gcd(m, b) = 1$ , and  $m' = -m^{-1} \bmod b$  comprising the steps of:

representing a value  $A$  in the Montgomery multiplication algorithm with a redundant notation; and

performing carry-save addition within an iterative process of the Montgomery multiplication algorithm to compute a new value of  $A$ , and further comprising executing the algorithm:

a.  $S \leftarrow 0$  and  $T \leftarrow 0$  (where  $S = (s_n s_{n-1} \dots s_1 s_0)_b$  and  $T = (t_n t_{n-1} \dots t_1 t_0)_b$ )

b. For  $i$  from 0 to  $(n-1)$ , do the following:

b1.  $u_i \leftarrow (s_0 + t_0 + x_i y_0) m' \bmod b$

b2.  $(S, T) \leftarrow (S + T + x_i y + u_i m) / b$

c.  $A_1 \leftarrow S + T$  and  $A_2 \leftarrow S + T - m$

d. If  $A_2 \geq 0$  then  $A \leftarrow A_2$  else  $A \leftarrow A_1$

e. Return  $A$ .

2. (cancelled)

3. (currently amended) The method of Claim [2] 1, wherein the value of  $b$  is two.

4. (currently amended) The method of Claim [2] 1, wherein step b2 of the algorithm further comprising selecting one of  $y+m$ ,  $y$ ,  $m$ , or 0 to be added to  $S$  and  $T$ .

5. (original) The method of Claim 1, wherein said representing step further comprises replacing said value of  $A$  with two other values, wherein a sum of the other two values equals said value of  $A$ .

6. (original) The method of Claim 1, wherein said performing step further comprises performing said carry-save addition in parallel.

7. (original) The method of Claim 1, wherein said performing step further comprises executing said iterative process a plurality of times equal to a number of bits representing said value of  $x$ .

8. (currently amended) A method of performing a Montgomery exponentiation of  $x^e \bmod m$  wherein the notation  $\text{Mont}(x, y, m)$  denotes the Montgomery multiplication algorithm  $yxR^{-1} \bmod m$  from input integers  $m = (m_{n-1} \dots m_1 m_0)_b$ ,  $x = (x_{n-1} \dots x_1 x_0)_b$ ,  $y = (y_{n-1} \dots y_1 y_0)_b$ , with  $0 \leq x, y < m$ ,  $R = b^n$  with  $\gcd(m, b) = 1$ , and  $m' = -m^{-1} \bmod b$  comprising the steps of:

representing a value  $A$  in the Montgomery multiplication algorithm with a redundant notation; and

performing carry-save addition within an iterative process of the Montgomery multiplication algorithm to compute a new value of  $A$ , wherein  $m$  has  $s$  bits,  $e$  has  $t+1$  bits, and  $0 < x < m$ , and the Montgomery exponentiation is performed according to the algorithm:

- a. Compute  $R = b^s$
- b. Compute  $R * R \bmod m$
- c. Compute  $z = \text{Mont}(x, R * R \bmod m, m)$
- d. Compute  $B = R \bmod m$
- e. For  $i$  from  $t$  to  $0$  do the following:
  - e.1  $B = \text{Mont}(B, B, m)$
  - e.2 If  $e_i = 1$  then  $B = \text{Mont}(B, z, m)$
- f.  $B = \text{Mont}(B, 1, m)$
- g. Return  $B$ .

9. (cancelled)
10. (original) The method of Claim 8, further comprising executing the algorithm for Mont ( $x, y, m$ ):
- a.  $S \leftarrow 0$  and  $T \leftarrow 0$  (where  $S = (s_n s_{n-1} \dots s_1 s_0)_b$  and  $T = (t_n t_{n-1} \dots t_1 t_0)_b$ .)
  - b. For  $i$  from 0 to  $(n-1)$ , do the following:
    - b1.  $u_i \leftarrow (s_0 + t_0 + x_i y_0) m' \bmod b$
    - b2.  $(S, T) \leftarrow (S + T + x_i y + u_i m) / b$
  - c.  $A_1 \leftarrow S + T$  and  $A_2 \leftarrow S + T - m$
  - d. If  $A_2 \geq 0$  then  $A \leftarrow A_2$  else  $A \leftarrow A_1$
  - e. Return  $A$
11. (original) The method of Claim 10, wherein the value of  $b$  is two.
12. (original) The method of Claim 10, wherein step b2 of the algorithm further comprising selecting one of  $y+m, y, m$ , or 0 to be added to  $S$  and  $T$ .
13. (original) The method of Claim 8, wherein said representing step further comprises replacing said value of  $A$  with two other values, wherein a sum of the other two values equals said value of  $A$ .
14. (original) The method of Claim 8, wherein said performing step further comprises performing said carry-save addition in parallel.
15. (original) The method of Claim 8, wherein said performing step further comprises executing said iterative process a plurality of times equal to a number of bits representing said value of  $x$ .
16. (original) The method of Claim 8, further comprising solving a plurality of Montgomery multiplication problems simultaneously.
17. (original) The method of Claim 16, and further comprising controlling selection of values to be added to  $S$  and  $T$  corresponding to each said of said plurality of Montgomery multiplication problems.

18. (original) An apparatus for performing a Montgomery multiplication for an output of  $xyR^{-1} \bmod m$  and an input of  $m = (m_{n-1} \dots m_1 m_0)_b$ ,  $x = (x_{n-1} \dots x_1 x_0)_b$ ,  $y = (y_{n-1} \dots y_1 y_0)_b$ , with  $0 \leq x, y < m$ ,  $R = b^n$  with  $\gcd(m, b) = 1$ , and  $m' = -m^{-1} \bmod b$ , comprising:

a plurality of storage units, with each one of the plurality of storage units corresponding to a value of  $x$ ,  $m$ ,  $y$ ,  $S$  and  $T$ , where  $A = S + T$ , wherein the corresponding storage unit for  $x$  can shift its value, and the corresponding storage units for  $S$  and  $T$  can be set to zero;

an adder configured to add the  $k$  terms from  $x_i$  times  $y$ , the  $k$  terms from  $u_i$  times  $m$ , and  $S$  and  $T$ ; and

a carry look-ahead adder configured to add any one set of values from a group including  $S$  and  $T$ , and  $S$  shifted and  $T$  shifted.

19. (original) The apparatus of Claim 18, wherein for  $b$  greater than or equal to two, the adder further comprises a carry-save adder within an iterative process of the Montgomery multiplication algorithm to compute a new value of  $A$ .

20. (original) The apparatus of Claim 19, wherein said carry-save adder further comprises a Wallace tree.

21. (original) The apparatus of Claim 18, wherein  $b$  is equal to two.

22. (original) The apparatus of Claim 21, wherein the adder further comprises a full adder configured to add input values of  $S$ ,  $T$ , and any one value from a group consisting of  $m$ ,  $y$ ,  $m+y$ , zero, and  $\sim m$  to produce a sum value and a carry value.

23. (original) The apparatus of Claim 18, wherein said plurality of storage units and said added are operatively coupled to execute the algorithm:

- a.  $S \leftarrow 0$  and  $T \leftarrow 0$  (where  $S = (s_n s_{n-1} \dots s_1 s_0)_b$  and  $T = (t_n t_{n-1} \dots t_1 t_0)_b$ .)
- b. For  $i$  from 0 to  $(n-1)$ , do the following:
  - b1.  $u_i \leftarrow (s_0 + t_0 + x_i y_0) m' \bmod b$
  - b2.  $(S, T) \leftarrow (S + T + x_i y + u_i m) / b$

24. (original) The apparatus of Claim 23, wherein said carry look-ahead adder is adapted to execute the algorithm:

- c.  $A_1 \leftarrow S + T$  and  $A_2 \leftarrow S + T - m$
- d. If  $A_2 \geq 0$  then  $A \leftarrow A_2$  else  $A \leftarrow A_1$

25. (original) The apparatus of Claim 23, wherein the storage units corresponding to said value of  $x$ ,  $y$ ,  $m$ ,  $y+m$ ,  $S$ , and  $T$  are further adapted to hold a plurality of Montgomery multiplication problems that are solved simultaneously, and further comprising means for controlling the selection of values to be added to  $S$  and  $T$  corresponding to each said of said plurality of Montgomery multiplication problems.

26. (original) The apparatus of Claim 25, wherein for each of said plurality of Montgomery multiply problems, the carry look-ahead adder is adapted to execute the algorithm:

- c.  $A_1 \leftarrow S + T$  and  $A_2 \leftarrow S + T - m$
- d. If  $A_2 \geq 0$ , then  $A \leftarrow A_2$ , else  $A \leftarrow A_1$ .